UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERIA

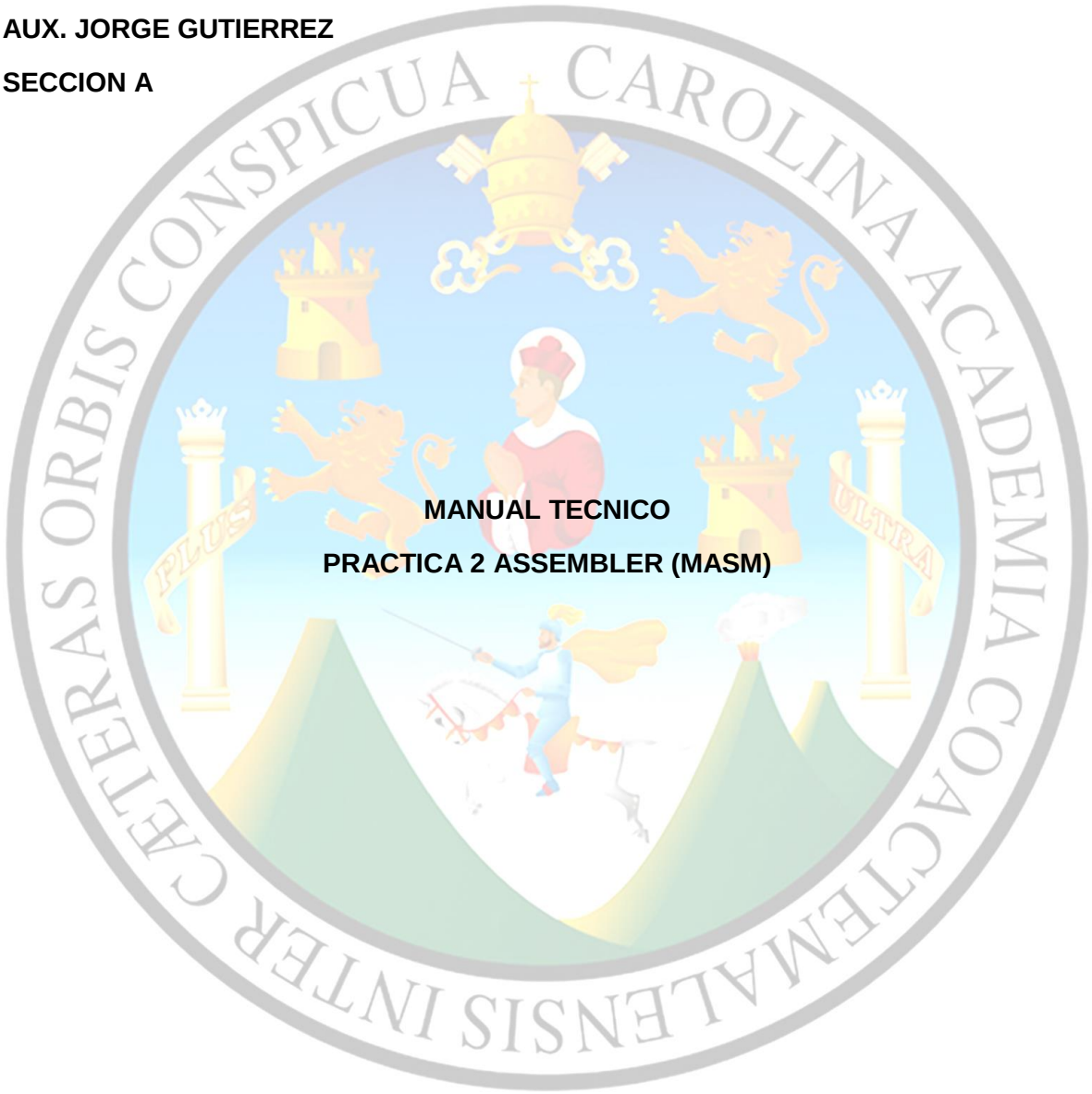
CIENCIAS Y SISTEMAS

ARQUITECTURA DE COMPUTADORES 1

ING. OTTO RENE LEIVA

AUX. JORGE GUTIERREZ

SECCION A



MANUAL TECNICO

PRACTICA 2 ASSEMBLER (MASM)

NOMBRE: OSCAR RENE CUELLAR MANCILLA

CARNET: 201503712

FECHA: 20 DE SEPTIEMBRE DE 2017

ENSAMBLADOR USADO EN LA PRACTICA: (MASM)

El Microsoft Macro Assembler (MASM) es un ensamblador para la familia x86 de microprocesadores. Fue producido originalmente por Microsoft para el trabajo de desarrollo en su sistema operativo MS-DOS, y fue durante cierto tiempo el ensamblador más popular disponible para ese sistema operativo. El MASM soportó una amplia variedad de facilidades para macros y programación estructurada, incluyendo construcciones de alto nivel para bucles, llamadas a procedimientos y alternación (por lo tanto, MASM es un ejemplo de un ensamblador de alto nivel). Versiones posteriores agregaron la capacidad de producir programas para los sistemas operativos Windows. MASM es una de las pocas herramientas de desarrollo de Microsoft para las cuales no había versiones separadas de 16 bits y 32 bits.

INTERRUPCIONES USADAS EN LA PRACTICA:

Interrupción al sistema 21H:

La mayoría de servicios ó funciones del sistema operativo MS-DOS se obtienen a través de la interrupción software 21H. Es por esto que se le denomina DOS-API: DOS-APPLICATION-PROGRAM-INTERFACE La INT 21H está compuesta por un grupo de funciones. Cuando se accede a la INT 21H, hay que indicar el número de función que queremos ejecutar. La llamada a la INT 21H se realizará como sigue:

- Introducimos en (AH) el número de función a la que deseamos acceder.

- En caso de que deseemos acceder a una sub-función dentro de una función, debemos indicarlo introduciendo en (AL) el número de esa sub-función.

- Llamar a la INT 21H.

CÓDIGO RELEVANTE DE LA PRACTICA:

Durante la realización de la práctica se utilizaron varios macros, el más utilizado sería el macro llamado “print” con el cual mando como parámetro una cadena a imprimir, se manda al registro AX el @data que representa que se va a escribir una cadena, con el mov ah,09 le indico a mi interrupción 21H que iniciare una impresión de cadena en pantalla y al registro DX le mando la dirección donde se almacena mi cadena a imprimir

```
print macro cadena
  mov ax,@data
  mov ds,ax
  mov ah,09
  mov dx,offset cadena
  int 21h
endm
```

También se crearon los macros para el manejo de los archivos, con nombres específicos que le dan un apego al lenguaje C.

```
fopen macro ruta,handle
print ruta
lea dx,ruta
mov ah,3dh
mov al, 00h
int 21h
mov handle,ax
jc NoSeEncontroRuta
endm

fwrite macro numbytes,databuffer,handle
mov ah,40h
mov bx,handle
mov cx,numbytes
lea dx,databuffer
int 21h
endm

fread macro numbytes,databuffer,handle
mov ah,3fh
mov bx,handle
mov cx,numbytes
lea dx,databuffer
int 21h
jc ErrorAlLeer
endm

fclose macro handle
mov ah,3eh
mov bx,handle
int 21h
jc ErrorAlCerrar
endm
```

En estos macros se maneja un handle, que es un número que representa de manera única a un archivo para poder ser manejado de una manera más accesible. Para entrar más a detalle acerca de los valores que reciben las interrupciones para poder realizar el manejo de archivos consultar la siguiente página donde se documenta que valores devuelve y pide cada función dada en el registro alto de AX.

Pagina:

http://ict.udlap.mx/people/oleg/docencia/ASSEMBLER/asm_interrup_21.html

Se declararon todos los mensajes que podían ser mostrados en consola con variables que los representaban de tipo byte:

```
;-------------------------MENSAJES-------------------------------------------------------
salt db 0ah,0dh, '  ','$'
;ENCABEZADO
enc0 db 0ah,0dh, '  UNIVERSIDAD DE SAN CARLOS DE GUATEMALA',0ah,0dh,'   FACULTAD DE INGENIERIA', 0ah,0dh, ' ESCUELA DE CIENCIAS Y SISTEMAS',0ah,0
;MENU PRINCIPAL
enc1 db 0ah,0dh, '  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%',0ah,0dh,'   %%%%%%% MENU PRINCIPAL %%%%%%%',0ah,0dh,'   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%',0ah,0
;RUTA DEL ARCHIVO
enc2 db 0ah,0dh, '  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%',0ah,0dh,'   %% INGRESE RUTA DEL ARCHIVO %%',0ah,0dh,'   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%','$'
;MENU DE OPERACIONES
enc3 db 0ah,0dh, '  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%',0ah,0dh,'   %%%% MENU DE OPERACIONES %%%%%',0ah,0dh,'   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%',0ah,0
enc4 db 0ah,0dh, '  REPORTE PRACTICA NO. 2',0ah,0dh,0ah,0dh,'   FECHA: 20 DE SEPTIEMBRE DE 2017','$'
enc5 db 0ah,0dh, '  HORA: ','$'
enc6 db 0ah,0dh, '  ENTRADA: ','$'
;MENSAJES DE ERRORES
err0 db 0ah,0dh, '  ERROR GENERAL, NO DEBERIA LLEGAR AQUI','$'
err1 db 0ah,0dh, '  EL NOMBRE DEL ARCHIVO NO TIENE EXTENSION','$'
err2 db 0ah,0dh, '  EXTENSION DEL ARCHIVO INCORRECTA',0ah,0dh,' INGRESE ARCHIVO CON EXTENSION .ARQ','$'
err3 db 0ah,0dh, '  EL NUMERO NO PUEDE CONTENER MAS DE 3 CARACTERES INCLUYENDO EL SIGNO','$'
err4 db 0ah,0dh, '  CARACTER INVALIDO, INGRESE NUMERO O SIGNO','$'
err5 db 0ah,0dh, '  OPERADOR INVALIDO, INGRESE (+,-,/,*)','$'
err6 db 0ah,0dh, '  NUMERO INVALIDO VUELVA A INTENTARLO FORMATO: (00,01,02,..08)','$'
err7 db 0ah,0dh, '  ERROR AL INTENTAR CREAR ARCHIVO','$'
err8 db 0ah,0dh, '  ERROR AL INTENTAR ABRIR ARCHIVO',0ah,0dh,'  PUEDE QUE EL ARCHIVO NO EXISTA','$'
err9 db 0ah,0dh, '  ERROR AL INTENTAR ESCRIBIR EN ARCHIVO','$'
err10 db 0ah,0dh, ' ERROR AL INTENTAR LEER EN ARCHIVO','$'
err11 db 0ah,0dh, ' ERROR AL INTENTAR CERRAR EL ARCHIVO','$'
err12 db 0ah,0dh, ' CARACTER INVALIDO SE ESPERABA NUMERO, VINO ->','$'
err13 db 0ah,0dh, ' CARACTER INVALIDO SE ESPERABA ESPACIO EN BLANCO, VINO ->','$'
err14 db 0ah,0dh, ' CARACTER INVALIDO SE ESPERABA SIGNO ARITMETICO, VINO ->','$'
err15 db 0ah,0dh, ' NO HA CARGADO NINGUNA RUTA PARA PODER GENERAR ',0ah,0dh,'   EL MISMO NOMBRE EN EL REPORTE','$'
err16 db 0ah,0dh, ' SE ESPERABA PUNTO Y COMA PARA FINALIZAR LA CADENA DEL ARCHIVO','$'
;MENSAJES PARA EL MODO CALCULADORA
cal0 db 0ah,0dh, '  NUMERO: ','$'
cal1 db          '  OPERADOR ARITMETICO: ','$'
cal2 db          '  RESULTADO: ','$'
cal3 db 0ah,0dh, '  ¿DESEA SALIR DE LA CALCULADORA?',0ah,0dh, ' 1. SI',0ah,0dh, '   2. NO','$'
;MENSAJES FACTORIAL
fac1 db 0ah,0dh, '  FACTORIAL: ','$'
```

Cada uno de los cuales es llamado en diferente ocasión y con diferente propósito.

Para obtener la ruta que ingresa el usuario desde la consola se utiliza la siguiente función:

```
;--------------OBTENER RUTA DEL ARCHIVO A CARGAR---------------
ObtenerRuta macro buffer
mov ah,01h
int 21h
cmp al,23h ;ascii del # ;omite todos los # que encuentre al escribir la ruta ejemplo: ##c:\entrada.arq##
je Obtener_Ruta
cmp al,0dh ;ascii del \n
je FinRuta
mov buffer[si],al
inc si
jmp Obtener_Ruta
FinRuta:
mov buffer[si],00h
xor si,si
endm
```

En ella leo carácter por carácter que se va ingresando, primero comparo si el carácter es igual a (#), si es igual lo omito y sigo leyendo los demás caracteres, luego si el carácter es un salto de línea es porque el usuario oprimió enter para finalizar el ingreso de la ruta entonces llamo al método FinRuta para salir del ciclo, si aún no se ha presionado enter, entonces guardo el carácter en un arreglo que me representa a mis caracteres de la ruta y aumento el registro de desplazamiento (si) en uno.

Búsqueda de errores en el archivo: para ella realizo el siguiente algoritmo:

```
;--------------VERIFICAR SI EL ARCHIVO DE ENTRADA CONTIENE ERRORES---------------------
VerificarErrores macro buffer
xor si,si
xor cx,cx
seguir:
;-----esperando numero
mov al,buffer[si]
cmp al,30h
jb Error12
cmp al,39h
ja Error12
inc si
inc cx
mov al,buffer[si]
cmp al,30h
jb Error12
cmp al,39h
ja Error12
inc si
inc cx
;-----puede venir punto y coma
mov al,buffer[si]
cmp al,3bh ;ascii del punto y coma
je TerminaVR
cmp al,24h ;ascii del $
je Error16
;-----esperando espacio en blanco
cmp al,20h ;ascii del espacio en blanco
jne Error13
inc si
inc cx
;-----esperando algun simbolo aritmetico
mov al,buffer[si]
cmp al,2ah ;ascii del *
je sigue
cmp al,2bh ;ascii del +
je sigue
cmp al,2dh ;ascii del -
je sigue
cmp al,2fh ;ascii del /
je sigue
jmp Error14
;------esperando espacio en blanco
sigue:
inc si
inc cx
mov al,buffer[si]
```

El algoritmo anterior pretende leer carácter por carácter el arreglo donde tengo la información del archivo de entrada guardada y realiza un pequeño scanner en el cual espera primero un Numero (##) compuesto por 2 dígitos seguido de un espacio en blanco (ascii 32), seguido de un operador aritmético y si lo que encuentra es un (;) es porque la lectura llego a su fin y no hubo ningún error, si encuentra el operador aritmético el proceso vuelve a repetirse hasta encontrar el punto y coma o hasta llenar el arreglo, si el análisis es pausado en algún momento se manda a llamar a las etiquetas que muestran el error y mostrando que fue lo que lo produjeron.

REQUERIMIENTOS MINIMOS:

Modo Calculadora:

En este modo se pide al usuario que ingrese un número el cual se guarda en una variable de tipo Word al igual que el segundo número, el operador se guarda en una variable de tipo byte, luego, en los macros se encargan de operar los dos números ingresados y guardar el resultado en otra variable de tipo Word.

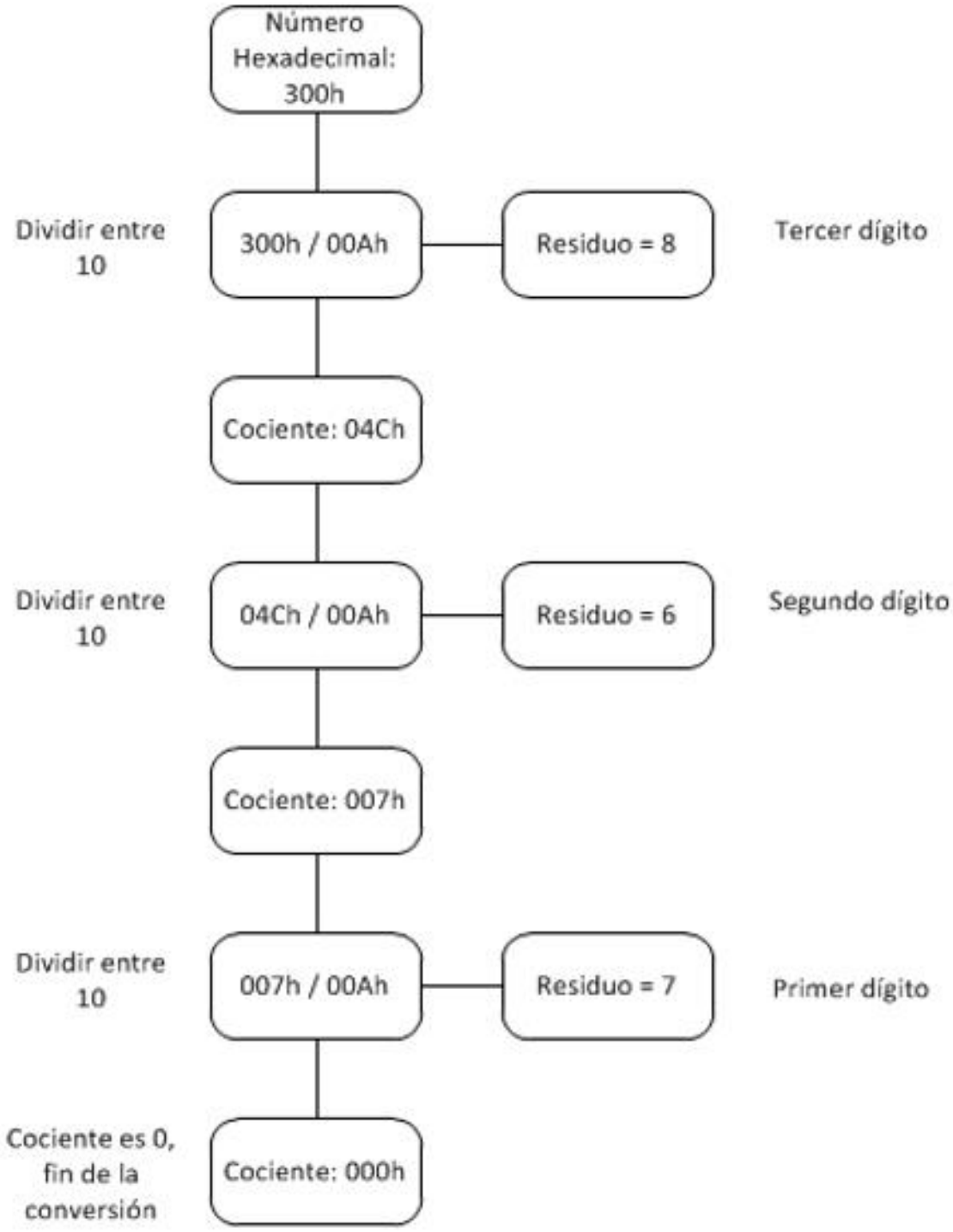
```
;--------------------MODO CALCULADORA-----------------------------------------------
    ModoCalculadora:
        LimpiarConsola
    ModoCalculadora1:
        LimpiarNumeros ArrayNum1,ArrayNum2,ArrayRes
        print cal0 ;Numero1
        ObtenerNumero1 ArrayNum1
        print cal1 ;Operador Aritmetico
        ObtenerOperador Operador
        print cal0 ;Numero2
        ObtenerNumero2 ArrayNum2                                        ;gua
        GuardarNumeros ArrayNum1,ArrayNum2,Numero1,Numero2,NegN1,NegN2,ANS,NegANS
        Operar Numero1,Operador,Numero2,Resultado,NegN1,NegN2,SignoMayor
        ConvertirResultado Resultado,ArrayRes                           ;con

        mov dx, Resultado
        mov ANS,dx                                                      ;gua
        print cal2 ;Resultado
        ImprimirSigno SignoMenos,NegN1,Operador,NegN2,NegRes,SignoMayor
        mov dx,NegRes
        mov NegANS,dx
        print ArrayRes
        print cal3  ;desea salir
        print salt

        getChar
        cmp al,31h
        je MenuPrincipal
        cmp al,32h
        je ModoCalculadora1
        jmp ModoCalculadora
```

Para poder mostrar el resultado en pantalla se utiliza el algoritmo de convertir un numero entero a sus respectivos asciis, les sumo 30h que es la diferencia que hay entre los números y sus códigos asciis.

Algoritmo:



Algoritmo implementado:

```
;---------------CONVERTIR RESULTADO--------------------
ConvertirResultado macro res,buffer
xor si,si
xor cx,cx
xor ax,ax
xor dx,dx
mov ax,res
mov dl,0ah
jmp segDiv

segDiv1:
xor ah,ah
segDiv:
div dl
;print msg10
inc cx
push ax
cmp al,00h ;si ya dio 0 en el cociente dejar de dividir
je FinCR
jmp segDiv1

FinCR:
pop ax
add ah,30h
mov buffer[si],ah
inc si
loop FinCR

mov ah,24h ;ascii del $
mov buffer[si],ah
inc si
endm
```

Analizar Archivo:

Otra parte de los requerimientos mínimos es el analizar el archivo y poder mostrar su resultado, para llevar esto a cabo se utilizan los siguientes algoritmos:

En el siguiente algoritmo se verifica que el archivo a cargar tenga la extensión valida, en este caso (.arq).

```
;-------------VERIFICAR QUE LA EXTENSION SEA .ARQ----------------------
VerificarExtension macro buffer
cmp buffer[si],2eh ;ascii punto
je VerSiguiente
inc si
inc cx
cmp cx,50 ;si ya es 50 no trae extension la ruta
je Error1
jmp Verificar_Extension
VerSiguiente:
inc si
cmp buffer[si],61h ;ascii letra a
jne Error2
inc si
cmp buffer[si],72h ;ascii letra r
jne Error2
inc si
cmp buffer[si],71h ;ascii letra q
jne Error2
xor si,si
endm
```

Para analizar lo que trae el archivo de entrada se utiliza la siguiente plantilla:

```
Num:
;------------GUARDANDO EL PRIMER NUMERO EN AL
xor ax,ax
xor bx,bx
xor dx,dx
mov dl,10
mov bl,bufferinfo[si] ;muevo el ascii del primer numero al registro BL
sub bl,30h ;le resto 30h que es la diferencia que hay entre los numeros
inc si ;paso al siguiente digito
add al,bl ;sumo lo que llevo en AL contra lo que acabo de encontrar
mul dl ;lo multiplico por 10
xor bl,bl
mov bl,bufferinfo[si] ;muevo el ascii del primer numero al registro BL
sub bl,30h ;le resto 30h que es la diferencia que hay entre los numeros
add al,bl ;sumo lo que llevo en AL contra lo que acabo de encontrar
;escribo el numero en mi buffer de sumas
mov buffersumas[di],ax
add di,02h ;aumento dos posiciones ya que es arreglo de tipo word
;print msg2
inc si ;antes numero, despues espacio en blanco o punto y coma
cmp bufferinfo[si],3bh ; ;
je FinAE
inc si ;despues signo aritmetico

Sim:
;-------------PREGUNTANDO QUE SIGNO VIENE
cmp bufferinfo[si],2bh ; +
je SYR
cmp bufferinfo[si],2dh ; -
je SYR
cmp bufferinfo[si],2ah ; *
je Opero
cmp bufferinfo[si],2fh ; /
je Opero
cmp bufferinfo[si],3bh ; ;
je FinAE
jmp Error
```

En el anterior algoritmo se ubican todas las divisiones y multiplicaciones que existan en el archivo, por ejemplo:

Entrada: 2 + 3 * 2 – 4 / 2 + 1;

En la siguiente entrada, el programa busca las multiplicaciones y divisiones y las opera, luego va escribiendo los resultados en un arreglo por aparte, como a continuación:

2 + 3 * 2 – 4 / 2 + 1; (opero 3*2 y 4/2) y cambio eso por su resultado.

2 + 6 – 2 + 1; (cambiado 3*2=6, 4/2=2)

Luego el arreglo se manda a otro macro que se encarga de realizar todas las sumas y restas.

Copio al registro AX el primer número, en este caso el (2), luego voy sumándole y restándole todo lo que traiga a la derecha.